**Tailored Application Access for Enhanced User Experience**

**1.Project Objectives**

This project, *Tailored Application Access for Enhanced User Experience*, is designed to address the challenge **Role-Based Access Optimization for Enhanced Efficiency**, is designed to address the challenge of streamlining user interactions within ServiceNow by delivering personalized and role-specific access. The goal is to implement a comprehensive solution leveraging advanced user-centric design principles, adaptive configuration tools, and intelligent access control mechanisms.

Through this initiative, we aim to enhance the user experience, boost operational efficiency, and fortify data security. This project aligns with the long-term vision of empowering employees and simplifying application interactions to create a seamless and productive digital environment within ServiceNow.

1. **Objectives**

**Business Goals**

1. **Improve Productivity for Each Role**:  
   Give employees access only to the applications and tools they need for their jobs, removing unnecessary distractions.
2. **Make the User Experience Simple**:  
   Clean up the ServiceNow interface by showing only relevant options, making it easier for employees to find what they need.
3. **Work Faster and Smarter**:  
   Help employee’s complete tasks quickly by reducing the time spent searching for the right tools and resources.
4. **Keep Data Safe**:  
   Use role-based access to protect important information and follow company security rules.
5. **Grow with the Organization**:  
   Build a system that can easily adapt to new roles, teams, and business needs as the company grows.

**Specific Outcomes**

1. **Custom Views for Each Role**:  
   Employees see only the menus and tools that are relevant to their job roles, making their work more focused and easier.
2. **Better Navigation**:  
   A simple and clear interface that helps employees work without confusion, improving their overall experience.
3. **Roles and Permissions Setup**:  
   Create specific roles and permissions for each team or department to ensure employees have access to the right tools.
4. **Organized Groups**:  
   Set up user groups and assign permissions based on roles for better control and teamwork.
5. **Save Time and Work Efficiently**:  
   Employees spend less time looking for tools, which boosts productivity across the company.
6. **Flexible System**:  
   Build a framework that can easily add new roles or tools without affecting current workflows.
7. **Strong Access Control**:  
   Ensure only authorized employees can access sensitive information, reducing security risks.
8. **Key features and Concepts Used**
9. **Role-Based Access:**Employees get access only to the tools they need for their job, keeping things simple and secure.
10. **Simplified User Interface:**A clean and easy-to-use interface that shows only the necessary options, helping employees find what they need quickly.
11. **Flexible Permissions:**You can easily set or change who has access to what, making it simple to manage user roles and permissions.
12. **Custom Menus for Each Role:**Menus and tools are customized for each employee based on their job, so they can focus on their tasks without distractions.
13. **Scalable System:**The system can grow as the company does, allowing new roles or tools to be added without disrupting current workflows.
14. **Detailed Steps to Solution Design**

**Pre-Requisites: -**

1. Knowledge on Service Now Administration.
2. Knowledge on Applications & Modules.

**Skills used to solve the problem statement: -**

1. Service Now Administration.

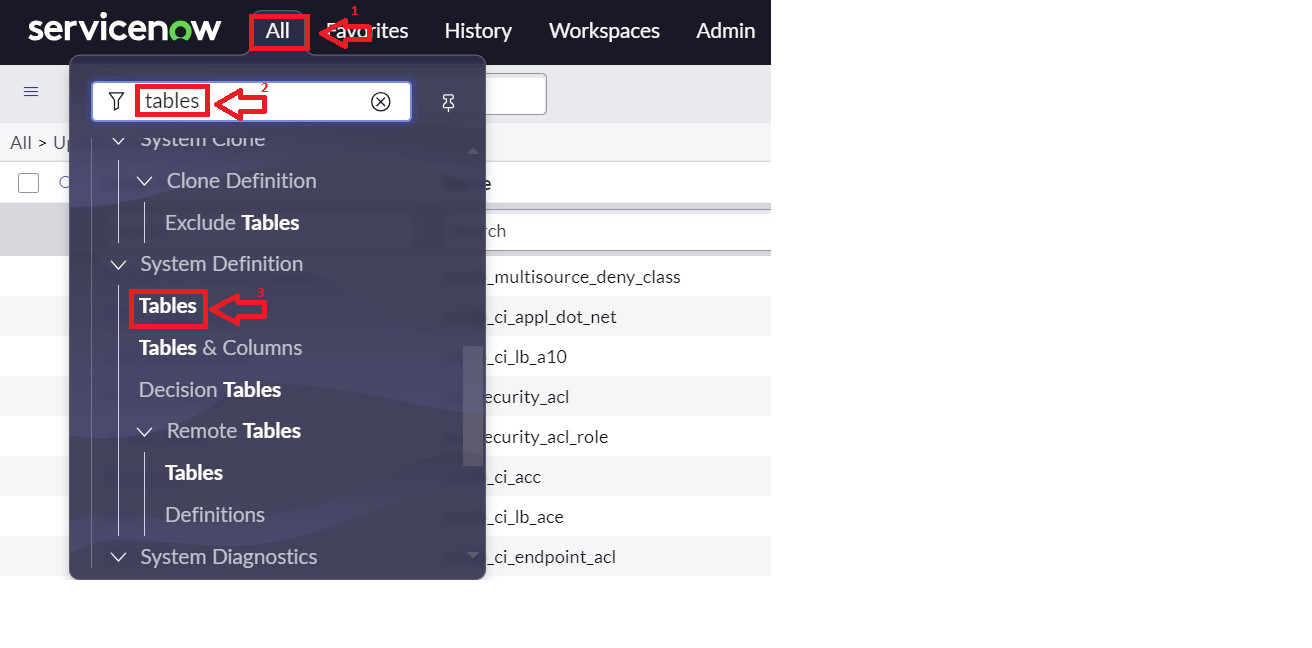
**Activity – 1:**

I started by opening the ServiceNow Developer Instance and clicked on "All" in the left-hand menu. Then, I searched for "Tables" in the search bar and selected "Tables" under the "System Definition" section. After that, I clicked on "New" to create a new table. I filled in the following details:

* **Label**: Service Request
* **Name**: Auto-Populated
* **Add module to menu**: Selected "Create New"
* I left everything else as the default settings.

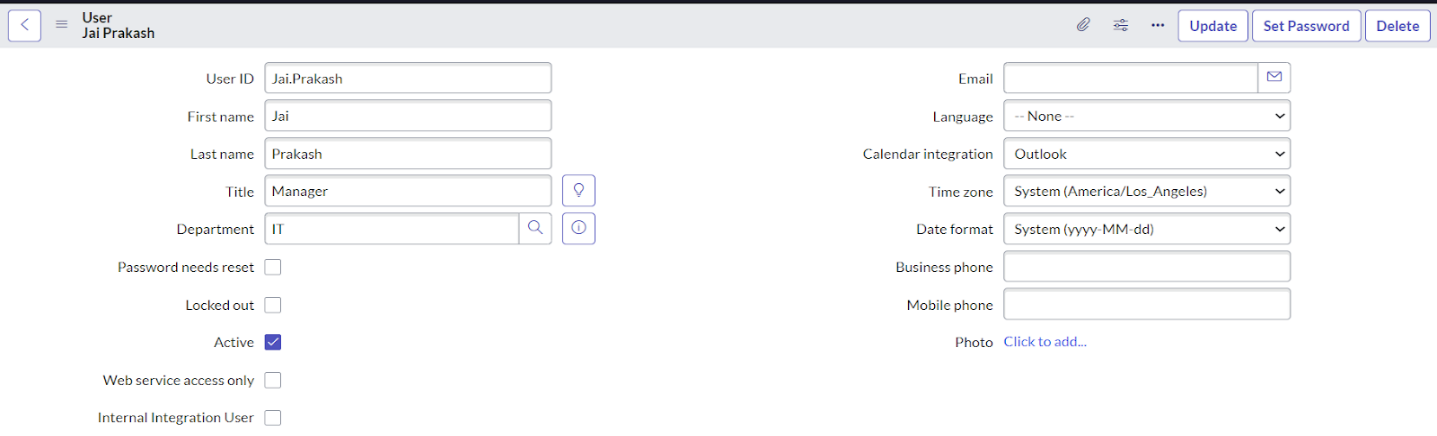
Next, under the "Columns" section, I clicked on "Insert a new row" and added the following columns:

* **Column label**: Name >> **Type**: String
* **Column label**: Issue >> **Type**: String

Finally, I clicked on "Submit" to create the table.

**Activity – 2:**

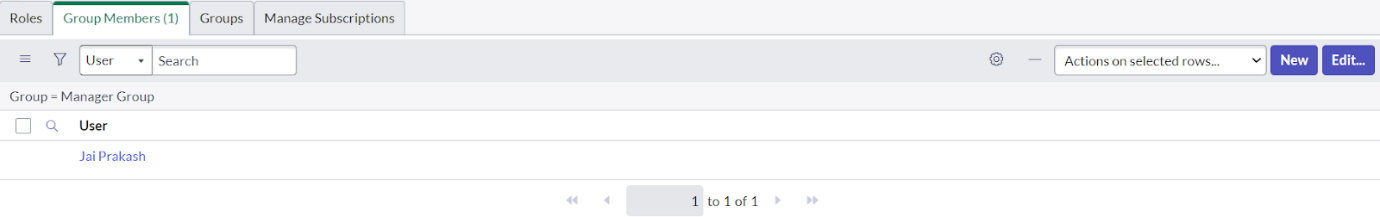
I opened ServiceNow and clicked on "All" in the left-hand menu. Then, I searched for "Users" in the search bar and selected "Users" under the "System Security" section. After that, I clicked on "New" to create a new user. I filled in the necessary details to create the user and then clicked on "Submit" to save the new user.



Activity – 3:

I opened ServiceNow and clicked on "All" in the left-hand menu. Then, I searched for "Groups" in the search bar and selected "Groups" under the "System Security" section. After that, I clicked on "New" to create a new group and filled in the required details.

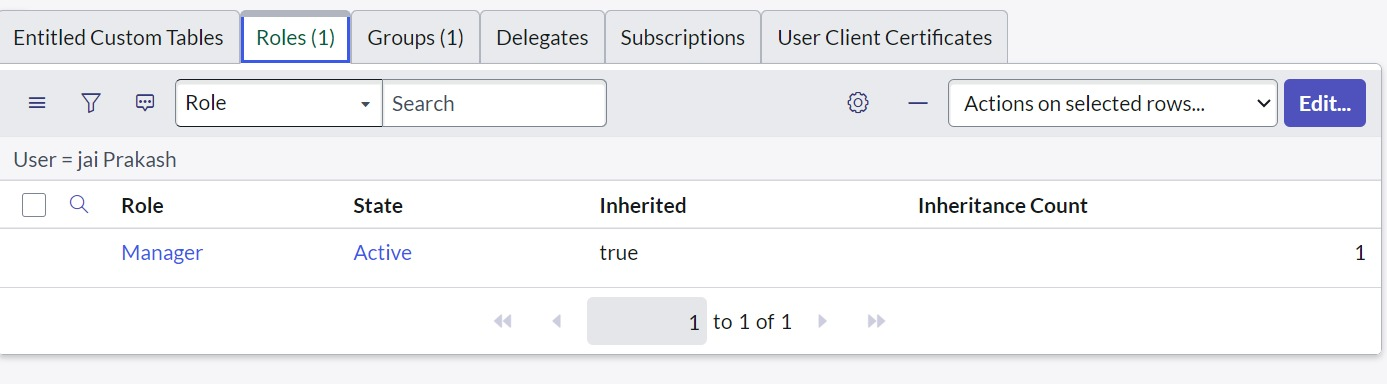
Next, under "Group Members," I clicked on "Edit" and added the user "Jai Prakash" to the Manager Group, then clicked on "Save." Finally, I clicked on "Save" again to save the group.



**Activity – 4:**

I opened ServiceNow and clicked on "All" in the left-hand menu. Then, I searched for "Roles" in the search bar and selected "Roles" under the "System Security" section. After that, I clicked on "New" to create a new role and filled in the required details. I then clicked on "Submit" to save the role.

Next, I clicked on "All" and searched for "Jai Prakash" under "Users." I opened the user record, went to the related list, and clicked on "Roles." I clicked on "Edit," added "Manager" to the selected list, and clicked on "Save" to apply the changes.



**Activity – 5:**

I clicked on "All" and searched for "Application Menus." Then, I opened the "Application Menus" section. Under the "Title," I searched for "Service Request" and opened it.

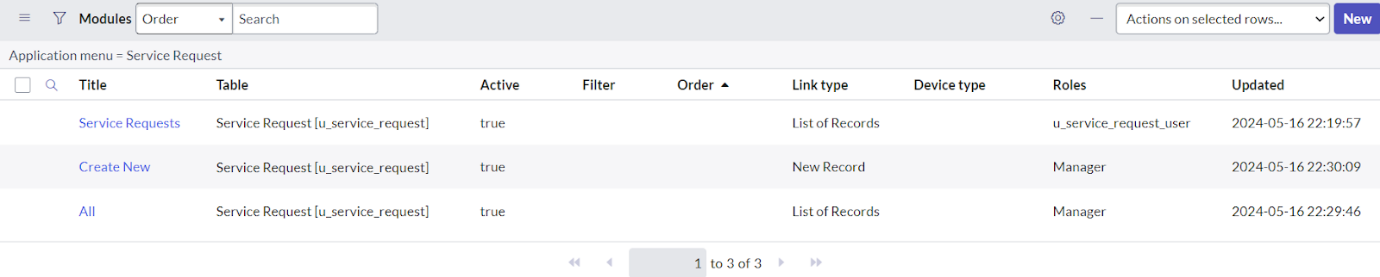
Under "Roles," I clicked on "Roles" and selected the role to which this should be viewed. After that, I clicked on "Done."

Next, under "Modules," I clicked on "New" and entered the following details:

* **Title**: Create New
* Under **Visibility**, I selected the role I wanted to assign.
* In **Link Type**, I filled in the details as shown in the figure.  
  Then, I clicked on "Save."

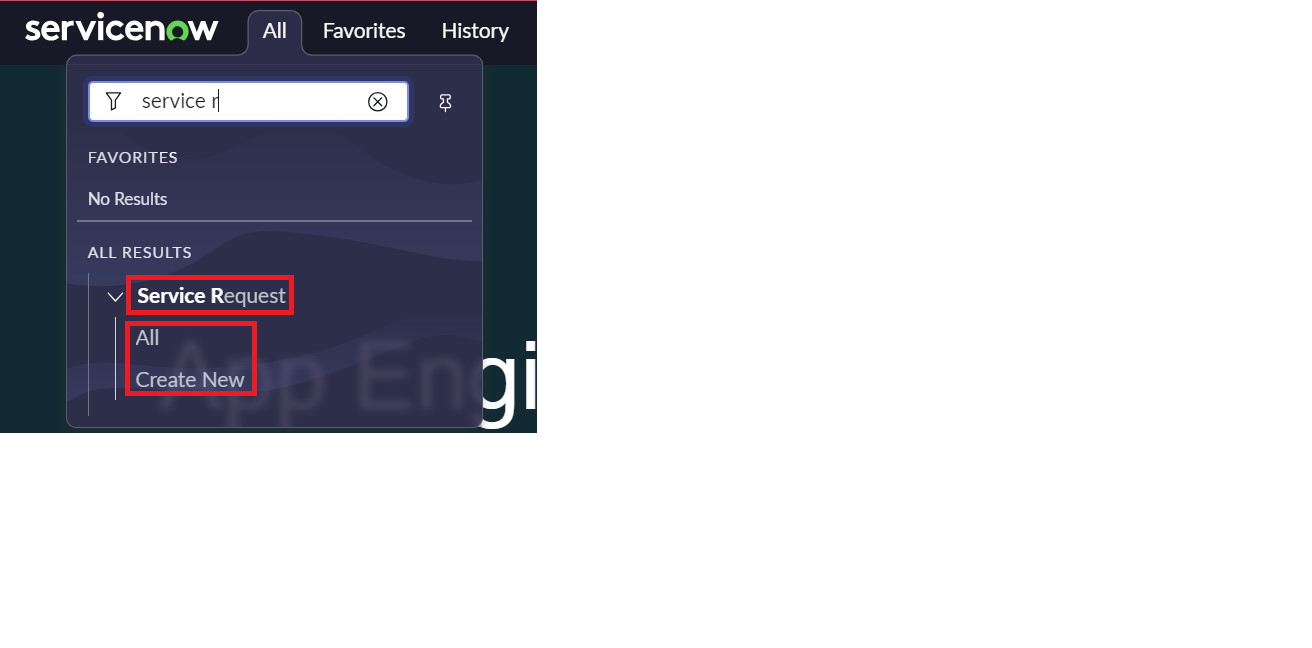
I repeated the process for another module with the following details:

* **Title**: All
* Under **Visibility**, I selected the role I wanted to assign.
* In **Link Type**, I filled in the details as shown in the figure.  
  After that, I clicked on "Save."



**Testing and Validation**

1. I went to Profile and clicked on Impersonate User.
2. I selected the user I had created and clicked on Impersonate User.
3. Then, I went to All and searched for Service Request.
4. I was able to find the application Service Request and the modules Create New and All



**Key Scenarios Addressed by ServiceNow in the Implementation Project:**

**Role-Based Access Control**: ServiceNow's role-based access control was implemented to ensure that employees can only access applications, modules, and data relevant to their specific job functions. This prevents unauthorized access and simplifies the user experience by eliminating unnecessary options.

**Streamlined User Management**: Through the creation of specific user roles and groups, ServiceNow enabled the management of users in a more organized manner. Each user is assigned appropriate permissions, ensuring they have access to only the resources needed for their tasks.

**Customized Service Request Management**: The application for **Service Request** was created, tailored to the organization’s needs. It allowed users to access the relevant modules (**Create New** and **All**) based on their role, streamlining the service request process.

**Efficient Role and Module Configuration**: By defining roles and modules such as **Create New** and **All**, ServiceNow allowed for the customization of the interface, ensuring that users see only the modules they need, improving workflow and reducing distractions.

**Scalable User and Group Management**: The system’s setup is designed to scale, allowing the addition of new roles, departments, or user groups in the future without disrupting current configurations. This ensures that ServiceNow will continue to meet the evolving needs of the organization.

**Improved Security and Compliance**: With the setup of custom roles and permissions, sensitive data is protected, and employees only have access to the information necessary for their tasks. This strengthens the organization's security measures and helps maintain compliance with internal policies.

Conclusion:

1. **Streamlined User Experience**: By creating specific roles and assigning relevant modules, employees now have access only to the applications and tools necessary for their job functions, reducing unnecessary distractions.
2. **Enhanced Access Control**: I implemented role-based access controls, ensuring that each user can only access the tools and data they need, thereby improving security and protecting sensitive information.
3. **Improved Workflow Efficiency**: The configuration of roles, groups, and modules has simplified the navigation and functionality of ServiceNow, allowing employees to find and use the required tools more quickly.
4. **Scalable Setup**: The structure I created is flexible and scalable, making it easy to add new roles, modules, or departments in the future without disrupting existing processes.
5. **Improved Productivity**: With streamlined access and reduced complexity, employees are now able to complete tasks more efficiently, leading to measurable improvements in overall productivity